Object-oriented Landmark Recognition for UAV-Navigation

E. Michaelsen, D. Roschkowski, L. Doktorski, K. Jaeger, M. Arens

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation
Gutleuthausstr. 176275 Ettlingen, Germany
eckart.michaelsen@iosb.fraunhofer.de

Abstract. Computer vision is an ever more important means for the navigation of UAVs. Here we propose a landmark recognition system looking for salient man-made infrastructure. An object-oriented structural system is preferred since it can utilize known properties of these objects such as part-of hierarchies, mutual geometric constraints of parts, generalization etc. The structure, available for use as landmark, will vary strongly with the region the UAV is supposed to navigate in. Clear object-oriented coding of the knowledge on the landmarks, their constraints, and their properties is a key to swift adaption. This contribution reports on an example: Adapting a system, designed for a central European country (Germany), for use in a more Eastern region (Turkey).

Introduction

Vision is an important sense in the field of autonomous robotics. Especially for navigation purposes it has been shown that vision can substantially enhance the abilities of autonomous mobile systems such as indoor robots or Unmanned Aerial Vehicles (UAVs) (compare [13]). Common approaches towards visual localization and mapping (V-SLAM) make use either of optical flow to estimate a short-term movement of the robot system, or orientate with respect to landmarks in the environment. These landmarks, in turn, can either be artificially placed in environment. This is possible under controlled circumstances and in small areas. Or, and this is the focus of this article, landmarks can be used for navigation, which are present in the environment anyway. For indoor robotics such landmarks could be doors, windows, etc. For the case of UAVs, man-made objects can be used as landmarks which exhibit a sufficient distinction with regard to their surrounding, such as highway bridges, highway crossings, or special purpose buildings such as churches or mosques.

Landmark recognition can be performed by means of a learning approach, meaning that one shows a systems several examples of, lets say, doors, and windows and the system comes up with a appearance based model of these objects. This approach can be taken in cases where landmarks exhibit not too much variation in appearance. This is not the case for landmarks such as highway bridges or mosques. Here, nevertheless, some common structural aspects of these objects can be described and exploited for landmark detection and usage, as we will show in the following.

Related Work

Structural knowledge-based computer vision in particular on aerial imagery has a remarkably long history [11]. Sophisticated production systems have been proposed e.g. in [2, 6]. Some of this work is being continued, including our own contributions cited below but often a certain lack of robustness was criticized. Attempts are known to unify statistical approaches with such syntactical or structural approaches [3]. It is difficult to achieve the low fault rates and high precision demanded by e.g., map update tasks in an open world where unknown objects will appear that have not been modeled yet. In a navigation control loop and fusion setting the requirements for error rates and precision can be set much lower without jeopardizing the overall robustness. Sometimes military UAVs and missiles already have automatic vision components included in their flight control. Emphasis is on real-time fusion of all available information sources - such as radar, GPS, INS, star trackers, GIS-data, altimeters, and also vision [1]. Because GPS may be subject to jamming, vision is regarded as valuable drift-less and precise source [4]. The civil side can learn a lot from the work published in that domain. Our own approach dates back more than 20 years [5, 12]. A recent renewal of this work assesses the structural approach for landmark-based UAV navigation, by closing a simulated control loop using Google-Earth as camera simulator and image source [10]. There are two keys to swift any-time performance and robustness of such systems: 1) the inclusion of clustering or accumulating productions [7] and 2) top down search rationales in addition to the quality driven interpretation included in the interpretation mechanism [8]. The production system approach can also be used for different recognition purposes such as finding building outlines for GIS-update based on gestalt relations [9].

Object-oriented landmark navigation

The regions of the world differ deeply in the kind of salient objects encountered there and in their frequency. An object-oriented system has to be flexible enough to load specific knowledge e.g., from ontologies about the area, where the UAV will be operating in. This is more promising than the use of learning pattern recognition techniques such as SVMs or statistical recognition based on image features which have been trained with non-representative data. Object recognition classes of our system are inherited from *CImageObject* as can be seen from the two example class diagrams in Figure 1. For each class, that has no decomposition link, a constructor is required that can segment such object from the image - most often this is a filter operation followed by a threshold. We call these classes primitive. Here we only have one primitive *CLine*, which results from a gradient filter. So these are small contour segments. These classes are the tools in a toolbox waiting for the user to choose a proper subset. Namely, for a certain landmark object, the flight planner picks the appropriate class from the box and also loads all specializations and parts.

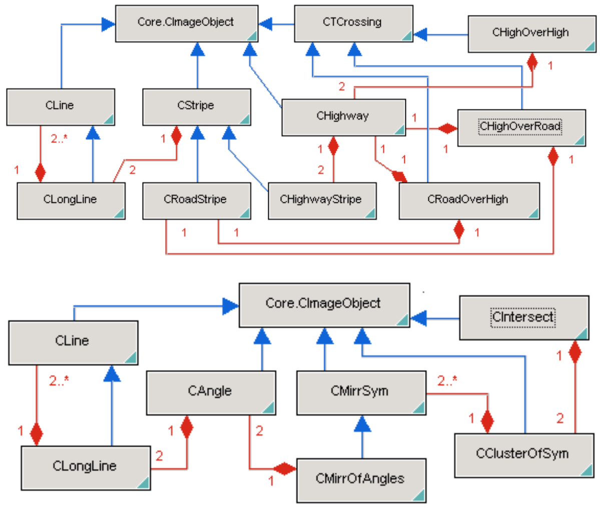

Fig. 1. Class diagrams of two recognition systems.

Running a Recognition with the System

The classes inherited from *CImageObject* constitute a production system. The robustness of such system can be properly assessed by using a Geo-System such as Google Earth as camera simulator in a flight control loop.

The accumulating interpreter: The first idea on the accumulative interpretation system has been published about twenty-five years ago [5]. Actually, the intended application and also the object classes where very similar to what is presented here. This work, however, was fairly preliminary. Interpreting uncertain data using such combinatorial descriptions is a non-trivial endeavor. For instance, following the part-of links of *CTCrossing* we see that each such object consists of six or eight CLongLine objects. From this we can infer that an exhaustive correct search will be of polynomial complexity of these orders in the number of *CLongLine* instances. Moreover, the number of primitives *CLine* in an instance of *CLongLine* has a lower bound of two but no upper bound. So here the interpretation has to search the power-set of the *CLine* instances which might be of exponential effort. Complete and sound interpretation - such as when using a PROLOG coding of the system - will usually not be feasible in the presence of about 20.000 primitive instances per image. Moreover, for a task like UAV navigation anytime performance is needed: The time window for making a course correction is limited. In a certain time interval an answer of the interpreter is required. There are means to overcome these problems mainly by avoiding exhaustive search and top-down control of the interpretation (see e.g. our own approach [8]). In particular the bounds 2..* in part-of links indicate clustering or Hough-like evidence accumulation which can be incorporated in the system by slight changes of the search algorithm [7]. Here we refer to [9, 12].

The Google Earth test-bed: Any recognition system has to be assessed with respect to the requirements of the task it is intended for. Accordingly, for vision based UAV navigation the goldstandard would be flying a real vehicle over the intended terrain and counting how often it runs astray. Since this may currently be hazardous, prohibited or quite expensive it should be simulated in an appropriate way. Internet-based Geo-Systems such as Google Earth provide an almost open world and a camera simulator which yields a picture for any given geo-coordinate. Different INS-drift models can be included. We used a simple Gaussian drift error where bias and standard deviation are growing linearly with the path length since the last update.

Experiments and Discussion

Fig. 2. Flight paths used in the experiments.

The Figure 2 shows the flight paths which we have used for the evaluation of our recognition system. It is evident that most of the experiments have been carried out using image data from Germany. Most flights consist of between 50 to 100 images. So the system has run on some thousand different images and proven robust. Instances of the class *CRoadOverHigh* can be found in Germany ubiquitously. Figure 3 shows a typical result for this model. This run is counted as success. The instances of the target class are displayed in white color.

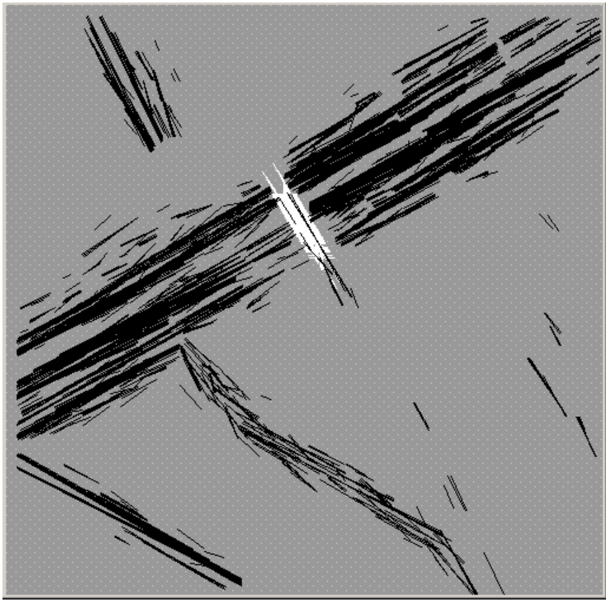

Fig. 3. A result for the class *CRoadOverHigh*.

In case of success, such instances cluster densely round the correct position. We also displayed in black color the *CLongLine* objects obtained in the given search-time. Although, the reader cannot reconstruct the image from this, he or she can assess the input characteristics of the system. Such complete success makes up roughly one third of the runs. One third is usually a complete failure - other objects mistaken for the target or nothing found at all - and the others are partially correct system answers still useful for navigation. That is good enough for almost always staying along the path and recovering from failure. For details we refer to [10]. The behavior of the system did not change much as we included parts of the neighboring countries Denmark and France into the experiments. Even for a path in the US, our system worked properly with the same unique landmark class. The same holds for two paths in north-western Turkey (Edirne to Istanbul and Istanbul to Ankara), however, for the rest of Turkey - and probably most of the rest of the world - landmarks fitting this class *CRoadOverHigh* are too rare. For the path further south from Cesme via Izmir to Aydin we had to extend the GUI for path planning so that it also includes a selection of different classes for each landmark. There are almost no bridges leading over the highway. Most of the Landmarks here are of the classes *CHighOverRoad* or *CHighOverHigh* and success rates seem a little lower but the system stays along the path and recovers from failure.

Recently, we took the classes from the lower diagram in Figure 1 and used them for our navigation system. Originally, these classes where designed for a complete different task - namely building outline recognition for GIS up-date from SAR images [9]. But the result displayed in Figure 4 shows that these classes can serve very well for landmark recognition. The intersection of perpendicular symmetry axes makes a very sharp and distinct landmark. The example building, we used here, is a mosque on the campus of the Marmara university, Istanbul. This landmark is part of the path from Istanbul along the south coast of the Marmara Sea which we currently use as test path for this kind of landmarks containing not only mosques but also other big symmetric buildings.

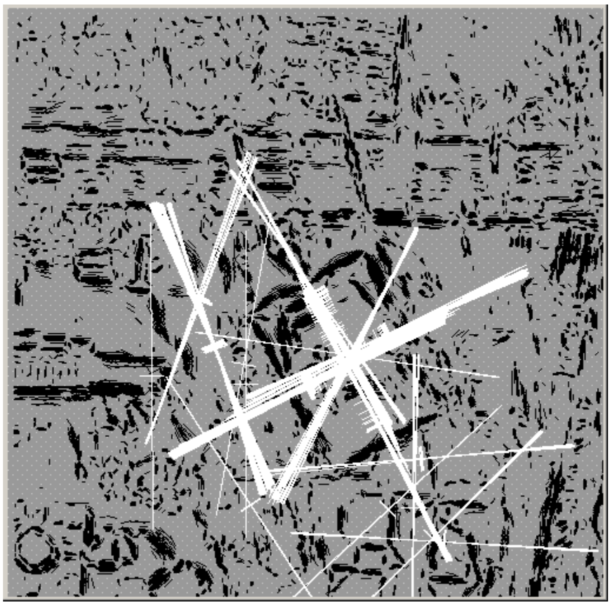

Fig. 4. A result for the class *CIntersect*.

We conclude that a structural landmark navigation system can be adapted to a different geographical scenario provided that the system is capable of easily including new knowledge in the form of classes suiting the new scenario. It is evident that in populated regions where landmarks of known structure and measures such as major highways are missing large salient buildings formed according to known common principles - such as mosques or churches – can be used as persistent geographical landmarks. While a learning system would require a new representative data set, a knowledge-based system requires new scene specific descriptive knowledge.

References

1. O. Aboutalib, B. Awalt, A. Fung, B. Thai, J. Leibs, T. J. Klausutis, R. Wehling, M. Iames: All source adaptive fusion for aided navigation in non-GPS environment. In: Z. Rahman, S. E. Reichenbach, M. A. Neifeld (eds.): SPIE XVI, Vol. 6575. 8-13, 2007.
2. B. Draper, R. Collins, J. Brolio, A. Hanson, E. Riseman. The Schema System. IJCV (2): 209-250, 1989.
3. C.-E. Guo, S. C. Zhu, Y. N. Wu: Modelling visual patterns by integrating descriptive and generative methods. International Journal on Computer Vision, 53 (1): 5-29, 2003.
4. Z. He, R. V. Iyer, P. Chandler: Vision-based UAV Flight Control and Obstacle Avoidance. IEEE American Control Conference, Minneapolis, MN, 2166 -2170, June 2006.
5. K. Luetjen: Ein Blackboard-basiertes Produktionssystem fuer die automatische Bildauswertung. In: G. Hatmann (ed.): Mustererkennung 1986, DAGM 1986. Informatik Fachberichte 125, Springer, Berlin: 164-168, 1986.
6. T. Matsuyama, V. S.-S. Hwang. Sigma a Knowledge-based Image Understanding System. Plenum Press, New York. 1990.
7. E. Michaelsen, L. Doktorski, M. Arens. Shortcuts in Production Systems - A Way to Include Clustering in Structural Pattern Recognition. PRIA-9 2008, vol. 2. : 30-38, Lobachevsky State University, Nizhni Novgorod, ISBN: 978-5-902390-14-5, 2008.
8. E. Michaelsen, M. Arens, L. Doktorski. Interaction of Control and Knowledge in a Structural Recognition System. In: Mertsching, B., Hund, M., Aziz, M. Z. (eds.) KI 2009, LNAI 5803, Springer, Berlin: 73-90, 2009.
9. E. Michaelsen, U. Stilla, U. Soergel, L. Doktorski. Extraction of building polygons from SAR images: Grouping and decision-level in the GESTALT system. Pattern Recognition Letters 31 (10): 1071-1076, 2010.
10. E. Michaelsen, K. Jaeger: A GOOGLE-Earth Based Test Bed for Structural Image-based UAV Navigation. 12th International Conference on Information Fusion, Seattle, WA, USA, July 6-9, 2009, IEEE-ISIF, Proc. on CD, ISBN: 978-0-9824438-0-4, pp. 340-346.
11. M. Nagao, T. Matsuyama. A Structural Analysis of complex Aerial Photographs. Plenum Press, New York, 1980.
12. U. Stilla and E. Michaelsen: Semantic Modelling of Man-made Objects by Production Nets. In: A. Gruen, E. P. Baltsavias, O. Henricsson (eds.): Automatic Extraction of Man-made Objects from Aerial and Space Images (II). Birkhaeuser, Basel: 43-52, 1997.
13. F. Bonin, A. Ortiz, and G. Oliver: Visual Navigation for Mobile Robots: A Survey. Journal of Intelligent and Robotic Systems 53:3(2008) 263-296.